

WORDEASE II

Wordease is a Text Editor/Processor written for Nascom 1/2. Wordease is intended to be used as an aid to the production of formatted printed documents.

The editor section is used to enter and modify the text to be printed, the editor is of the full page type, is very easy to use, and has a number of powerful features. The editor also allows text to be dumped and loaded from tape.

The processor section is used to format and print the text. The processor interprets control commands embedded in the text (these are not printed) and formats the text accordingly.

Wordease occupies almost 5k of memory starting at X'1000'. Wordease should be executed at X'1000'. The area from X'0F00' to X'1000' is used for stack space and may not be used by other programs. The memory used for the text buffer normally starts at the end of the program and extends upwards to the limit set by the user, to redefine the text buffer and set the upper limit see appendix B.

Note - throughout this manual keyboard keys are referred to by certain names such as FIND, PAUSE, RETURN, etc. The actual key on your keyboard to which these names correspond depends on the system monitor and keyboard you are using, (see appendix A) in any case it is a simple matter to redefine these keys, appendix B gives details.

COPYRIGHT NOTE

This manual and Wordease are copyrighted. If you think this is a good program, worth what you paid for it (you did pay for it didn't you?), and would like to see more of the same then please don't distribute copies to friends etc.

THE EDITOR.

The Editor is of the full page type with full cursor control. The Nascom screen is split into three sections as follows:-

The top line which is used for a header, a bytes left display, and flashing message area.

The bottom line which is used as the command line.

The fourteen lines in between which can be regarded as a 'window' into the text buffer. The eighth line up from the bottom is considered to be the 'current' line for the MARK, EDIT, and COPY commands and is indicated by a flashing prompt. The first position of each line is reserved for markers set by the MARK command.

The cursor is normally a flashing 'hash' mark (code X'7F'), in insert mode it changes to a flashing 'pusher' (code X'17').

If the users system has a bell or buzzer (see Appendix B) it will bleep when error messages are output, or when an attempt to move the cursor off the screen is made.

Various control keys are used to enter and manipulate text in the text buffer as follows:-

CURSOR UP	Moves cursor up 1 line, if at top of screen then scrolls text down.
CURSOR DOWN	Moves cursor down 1 line, if at bottom of screen then scrolls text up.
CURSOR LEFT	Moves cursor left 1 character, wraps from start of line to end of last.
CURSOR RIGHT	Moves cursor right 1 character, at end of line scrolls text above up 1 line and wraps to the start of a new blank line.
BACKSPACE	As cursor left but deletes character to the left of the cursor.

SCROLL UP	Scrolls text up 1 line.
SCROLL DOWN	Scrolls text down 1 line.
DELETE	Deletes the character at the cursor.
INSERT	Toggles the insert mode. (note cursor up, cursor down, and return reset insert mode).
FIND	This key is used to find, or find and replace the next string as defined by the FIND command. See FIND command for more detail.
CONTROL	This key has the same effect as typing an up arrow ie ↑ (the character used to indicate control commands in the processor).
RETURN	Return normally causes the text above the cursor to scroll up 1 line and the cursor is placed at the start of a new blank line. If the cursor is on the command line or return is hit twice then any commands on the command line will be executed.

Any other key will cause the appropriate character to be placed at the cursor position and the cursor moves right one place, again if at the end of a line the text will scroll up and a new blank line is started. In insert mode the text to the right of the cursor is pushed along to make room for the new character. If any text is pushed over the end of the line the text below is scrolled down and a new line started so no text will be lost.

Each line of text on the screen is stored in the text buffer with a maximum of one trailing space. Blank lines are ignored. Words which span across two lines remain intact, however, this is not recommended as subsequent insertions into the line will cause the word to be split.

The cursor control keys may also be used on the command line, however in insert mode text pushed over the end of the line will be lost.

Various commands are also used to manipulate the text in the buffer. These commands are entered on the command line after the prompt character in the bottom left hand corner. All commands except the find command may be stacked, that is entered in a string and executed in sequence. Command strings are terminated by a space and commands may optionally be separated by commas. On completion of a command string the command line is blanked and the cursor placed to the right of the prompt. If an unknown command is encountered a message is displayed and the cursor placed at the unknown command for correction. This feature is also useful for repeating command strings, if an invalid command (e.g. full stop) is placed at the end of a command string the command string may be repeated simply by hitting return.

Commands may be entered in upper or lower case and are as follows:-

- Un UP - Scrolls text up n lines. n is a decimal number in the range 0 - 255, if no number is entered then the default value is 1.
- Dn DOWN - Scrolls text down n lines.
- T TOP - Displays top of text buffer.
- B BOTTOM - Displays bottom of text buffer.
- L LOCK - This command is a form of shift lock, it causes lower case alphabetic characters to be forced to upper case and upper case to lower. The LOCK command turns this feature on and off alternately.
- E EDIT - Places the cursor at the start of the current line. It must be the only or last command of a string.
- F FIND - Sets find, or find and replace strings then finds first.

The FIND command may not be stacked with other commands. To set a find string simply follow the 'F' command with the string to find. If the first and last characters of the string are the same they are considered to be delimiters, delimiters are optional for find strings unless the string actually starts and ends with the same character.

To set find and replace strings delimiters (which may be any character not contained within the strings) must be used. Delimiters are required at the start of the find string, at the end of the replace string, and between the two.

Find and replace strings remain defined until redefined by another FIND command, setting a find string alone will delete the previous replace string.

When the strings have been defined or when 'F' with no string is entered the command starts to search for the first occurrence of the find string in the text buffer, from the current line on the screen downwards. When found the string is displayed on the current line along with the text buffer surrounding it, the cursor is placed at the start of the string, ready for modification. At this point if no replace string is defined the command has finished and normal editing is resumed. The search may be restarted at any time either by entering another FIND command (with no string) or preferably by hitting the FIND key, in this manner the text buffer may be scanned through for all occurrences of a particular string very quickly. If the string is not found a message is displayed and the cursor returned to the command line.

Alternatively if a replace string has been defined the editor will prompt with a 'REPLACE ? (Y/N)' message and wait for the keyboard, if FIND is pressed the next string will be searched for, if 'Y' is pressed the string will be replaced (see below for details), any other key will terminate the command.

NOTE - The find and replace string lengths are limited only by the length of the command line. The replace string may be longer, the same, or shorter (even null) than the find string. If the replace string is shorter then extra characters of the find string are deleted, if longer the extra characters are inserted, (as if the user had typed them in insert mode) text will not be lost.

The find and replace strings may also contain 'wild' characters represented by '*'. In the find string any text character will match a wild character. In the replace string wild characters are skipped over while replacing.

The FIND command can not find strings which span across two lines.

- M MARK - Marks the current line to define a block of text to be used by the SAVE, KILL, and COPY commands.

The first time a line is marked it is pointed at by a right hand arrow in the first position of the line, it is now defined as the text block. If another line is now marked an up or down arrow is placed in the first position to point at the line last marked, the other marking arrow is also changed to point towards this new line, this defines the text between the markers including the marked lines as the text block. If another line is now marked the least recent marker is deleted and the block is redefined between this line and that most recently marked. To define a single line again it is simply marked twice. To define the whole text buffer the command string 'TMBM' may be used.

- S SAVE - Causes the defined block of text to be saved on tape.

If no block is defined a marker error will be displayed and the command aborted. The tape must be started prior to the command.

The block is written to tape in a format allowing it to be read back into the text buffer simply by starting the tape. The cursor should be positioned at the place where the text is to be inserted, (hitting return to open a new line if required). The block is written with a number of leading nulls which stop the cursor flashing, each line has a small delay to allow the text to scroll, and the block is finished with two returns which put the cursor back on the command line. Characters are written to tape with even parity, received characters with bad parity are forced to '?' to aid error correction. Note that the LOCK command also operates on input from tape and must be turned off before reading from tape.

- R READ - Reads a block of text from tape. The command places the cursor on the current line and starts the tape, the tape is automatically stopped at the end of the block. The command must be the last or only command of a string. Any text on the current line will be overwritten, entering an EDIT command followed by two returns will open a new blank line to prevent this.

KILL KILL - Deletes the defined block of text.
'KILL' must be entered to avoid unintentional 'killing'. The block markers are also deleted. To clear the whole text buffer the command string 'TMBMKILL' may be used.

C COPY - Copies the defined block of text to below the current line.

A marker error message will be displayed if the current line is within the defined block and the command aborted. If there is not enough room in the buffer an overflow message will be displayed and the command aborted. The block markers are unchanged. To move a block of text the command string 'CKILL' may be used.

A ADJUST - Causes the text in the text buffer to be adjusted so that as many words as possible are placed on each line, words which were spread across two lines are reassembled. The command also recognises the processing commands NEW LINE, EJECT, and FINISH and the text following these commands will start on a new editor line. The command allows text to be typed into the editor as one long line which can then be adjusted to give a more readable format. Note the ADJUST command uses the screen as a buffer.

P PROCESS - Causes the text buffer to be processed and output to the defined ports, see PROCESS section.

Q QUIT - Returns the user to the system monitor.
Reset should not be used as the Editor must first close a playground area within the text buffer and delete all block markers.

GENERAL NOTES :-

Buffer overflow - The bytes free display on the top line of the screen gives an indication of space left in the text buffer. At all times 400 bytes are reserved for a playground area within the buffer, when there is no room for a new playground an overflow warning message is output, and the user should not enter more text as unpredictable results will occur. However depending on the amount of text on the screen there is usually enough room for one or two more lines (at your own risk).

THE PROCESSOR.

The processor section outputs the text buffer to defined ports after formatting according to control commands embedded in the text. All control commands are indicated by a preceeding up arrow ie ↑ (code X'5E'), in the following text n is used to represent a decimal number in the range 0 to 255, if no number is entered then the default value is 1.

Three output ports are catered for each with the following options:-

- 1 Character delay.
- 2 Use Nascom 1 CR code.
- 3 Inhibit line feeds.
- 4 Force upper case.

To set these options and output port vectors see Appendix B. It is recommended that port 1 be directed to the Nascom CRT routines for display of formatted text prior to printing.

The Process command 'P'

The process command may be followed by a number of parameters, these parameters may be entered in any order, and must not contain embedded spaces. The parameters may be entered in upper or lower case and are as follows:-

- 1,2,3 Directs output to ports 1,2, and 3. multiple ports may be set, processing still takes place when no ports are set, this may be used for error checking prior to printing.
- Sn Sets the starting page number to n, default is 1.
- Dn Sets the character delay value to n, default is 1. A value of 50 gives around 10 characters/second.
- Pn-n Selects pages to be printed, only pages with numbers between n and n inclusive will be output. e.g. a P1P3-5 command would cause only pages 3,4, and 5 of the text to be output to port 1.

Text from the text buffer is processed as a continuous string with no regard to the start of new lines as displayed by the editor. Normally all lines of text are justified to the defined line length (by inserting spaces), see below for

exceptions. During processing various errors can occur, in all cases the process command is aborted, a message displayed indicating the type of error, and the text buffer is redisplayed with the line on which the error occurred as the current line.

Once processing has started the command may be aborted by hitting the escape key. Processing may also be halted by hitting the pause key, hit any other key to continue.

Formatting Control Commands.

These commands may be entered in upper or lower case and are as follows:-

- ↑Ln set Line length - Sets the number of characters per line to n. An error will occur if the length of the current line being processed already exceeds n.
- ↑Pn set Page size - Sets the number of lines per page to n. An error will occur if the current page already has more than n lines.
- ↑D set Double space - Sets the double space mode in which each line of text is followed by a blank line. This command is useful for proof copies.
- ↑S set Single space - Resets the double space mode.
- ↑Jn Justify - Turns justification on and off. If n is 1 (or omitted) justification is turned on, if 0 it is turned off.
- ↑In set Indent - Causes an immediate tab to column n, and similar tabs at the start of all subsequent lines.
- ↑Tn Tab - Causes a tab to column n, if the current line has text at column n it will be overwritten. An error will occur if n exceeds the current line length value.
- ↑Nn New line - Causes the current line of text to be output without justification followed by n-1 new lines. If double space mode is set then 1 extra new line is output. If the new lines cause the end of page to be reached then no further new lines are output on the new page.

- ↑E Eject - Outputs new lines until a new page is started.
- ↑↑ up arrow - Causes an up arrow ↑ to be printed.
- ↑Un User - Allows the user to send special control characters to the printer (turn on underline, bell, etc). The decimal value of the required code must follow the command.
- ↑Cn Conditional eject - If there are less than n lines left on the current page then an eject will take place. This command is useful to avoid printing section titles etc as the last line of a page.
- ↑F Finish - This command is intended for use at the end of the text, it causes new lines to be output till the end of the current page is reached, no new page is started.
- ↑X output page number - Causes a three digit page number with leading zero suppression to be output. The maximum page number is 255.
- ↑H Halt - Causes the processor to wait for any key to be pressed, it is useful at the end of each page to allow the printer to be fed with a new sheet of paper. The processor will not halt if no text is being output.
- ↑0-9 macros - These commands are used to define and output macros 0-9. A macro is a section of text which may be defined once and subsequently output in several places.
To define a macro the command must be directly followed by the macro text enclosed in double quotes. Macros may be of any length and must be defined before use, they may subsequently be redefined. Macros may be nested, that is contain other macros, they may also contain any control commands, they may not contain macro definitions. If the command is not followed by a quoted string then the macro is output.
Macros 0 and 1 are special cases, they are automatically output at the start and end of each page. They may be used for headings and footings (particularly for page numbering).

Notes on the processor.

The descriptions of the format control commands above is fairly straight forward, however, there are a few points which merit further discussion.

Spaces - the format in which the text is entered in the editor generally has no effect on how it is processed, however it is important to remember that each line of text in the editor normally has a trailing space, these spaces are required to separate the last and first words of two lines. Sometimes these spaces cause problems, for instance if the user wishes to start a new paragraph after an indented section of text. The indented section may be finished with `↑N2↑IO` commands and for clarity the new paragraph started on a new editor line, this will not give the required result as the first character of the new paragraph will be the space which follows the `↑IO` command. To overcome this problem the `↑N` command (also `↑E`, `↑C`, `↑F`) removes following spaces and for this to work the `↑N2` must follow the `↑IO` command ie `↑IO↑N2`.

Indented sections - The indent command provides a means of indenting sections of text and is quite straight forward. Text may also be 'right indented' simply by shortening the line length with the `↑L` command, several examples appear in this manual. In the section which describes the formatting control commands you will notice each command is given with a block of indented text which describes it to the right. This can be achieved as follows:-

`↑IOcommand↑Il5`Start of description.

Numbers - if the text to be processed contains numbers and they directly follow a formatting command the processor may be confused and interpret the number as part of the command. This is easily avoided by leaving a space between the command and the number (tab and indent commands may need adjustment to allow for this space).

Macros - macros are a very powerful feature. Macros 0 and 1 are special cases discussed below. A particular use of macros is in the processing of standard letters where for each version only parts of the text such as the date, names, prices, etc require modification. These sections may be processed as macros which are defined at the begining of the text and as such are easily modified before printing.

Page sizes, top of page and bottom of page macros - to make Wordease more versatile no automatic title, page numbering, or end of page spacing features are included. All these features and more are available through the use of the special macros 0 and 1 (top of page and bottom of page). These macros are automatically output at the top and bottom of each page. Generally they will contain a number of new lines, possibly a title and page number command, and macro 1 will often contain a halt command.

The macros used to produce this manual are given below.

↑0"↑S↑IO↑L60↑N3↑2 User Manual↑N4"

↑1"↑S↑IO↑L60↑N3Copyright T.R.Harris/Program Power
1980↑T53Page ↑X↑N3↑P66↑H"

↑2"Wordease"

There are several points to note about these macros.

Notice that both macros set single space mode, no indent, and reset the line length, this is so that if a new page is started in the middle of an indented section the text within the macro will not be indented. This applies to the line length and space mode commands as well, these formatting variables changed within the macro will be restored to their original value at the end of the macro. Both macros contain new line commands before and after the text, this is to place the text in the desired position and to space over the perforations in fanfold stationery. Notice macro 2 is defined as "Wordease" and is used within macro 0, this was done so that an alternative name for this program could be easily substituted. Macro 1 contains a halt command which stops the processor at the end of the page and allows a new sheet of paper to be fed to the printer.

You may have noticed that macro 1 also contains a page size command, this is part of a solution to a problem we have not yet discussed. Getting the text in the right place on the page is always a problem with word processors. The trouble is that until the processor has output the bottom of page macro it can not know how many lines it contains, therefore it can not know when the macro should be output in the first place. For this reason as far as counting the number of lines on the page is concerned the processor considers the bottom of page macro to be part of the next page. Unfortunately this means that the first page to be printed will be too long by the number of lines in the bottom of page macro. There are two solutions to this problem, for example suppose the bottom of page macro

contains six lines, then at the beginning of the text we could place a ↑N6 command and ensure that the printer is set up with the start of the page six lines below the print head. A better solution is to initially define the page size to be six lines less than that actually required then redefine the page size within the bottom line macro so that all subsequent pages are of the correct size.

Another point to note is that the top of page macro is not automatically output at the start of the first page, if it is required a ↑0 command must be used at the start of the text (after the macro has been defined).

APPENDIX A

List of special function editor keys for NAS SYS and T4 monitors.

Name	NAS SYS	T4
RETURN	NEW LINE	NEW LINE
CURSOR UP	CURSOR UP	@U
CURSOR DOWN	CURSOR DOWN	@D
CURSOR RIGHT	CURSOR RIGHT	@R
CURSOR LEFT	CURSOR LEFT	@L
INSERT	SHIFT CURSOR RIGHT	@I
DELETE	SHIFT CURSOR LEFT	@K (kill)
SCROLL UP	CONTROL A	@A
SCROLL DOWN	CONTROL B	@B
BACKSPACE	BACKSPACE	BACKSPACE
FIND	CONTROL F	@F
CONTROL	CONTROL C	@C
ESCAPE	ESC	@E
PAUSE	SPACE	SPACE

The user may wish to redefine these keys especially if using a non standard keyboard, Appendix B gives details.

APPENDIX B

Wordease is very monitor independent. All communication with the system monitor takes place through the vectors described below. All external routines may use any registers. Wordease makes some use of the alternate register set but otherwise can run under interrupts.

X'1003'	PORT1	Vector to output port 1 routine, character to print in A.
X'1006'	PORT10	Options byte for port 1. This byte contains flags relevant to port 1 as follows Bit 0 (01) character delay option Bit 1 (02) Use Nascom 1 CR code option Bit 2 (04) Inhibit line feed option Bit 3 (08) Force upper case option These flags may be combined, eg a value of 05 would give character delay and no line feeds.
X'1007'	PORT2	As above
X'100A'	PORT20	As above
X'100B'	PORT3	As above
X'100E'	PORT30	AS above
X'100F'	KBD	Vector to keyboard input routine (not tape input aswell). Must return with carry if key is pressed and character in A.
X'1012'	TAPOUT	Vector to tape output routine. Character in A.
X'1015'	TAPEIN	Vector to tape input routine. Must return with carry if character received and character in A.
X'1018'	MOTFLP	Vector to routine which starts and stops tape transport.

X'101B' EXIT Vector to the monitor reentry point. Six bytes are reserved for this vector to allow for a call or restart to clear the Nascom screen before reentering the monitor.

X'1021' BLEEP Vector to a bleep routine to sound a system buzzer or bell if fitted.

All the above vectors contain jumps or restarts (for NAS SYS). Unused vectors eg BLEEP must be patched out with a return instruction.

X'1024' KTAB This is a table of key codes for the editor special function keys as follows:-

RETURN
CURSOR UP
CURSOR DOWN
CURSOR RIGHT
CURSOR LEFT
INSERT
SCROLL UP
SCROLL DOWN
DELETE
BACKSPACE
FIND
CONTROL

X'1030' SPEED The value of this byte determines the cursor blink rate. A value of 8 is suitable at 4MHz with normal keyboard routines.

X'1031' ESCAPE Code for escape key.

X'1032' PAUSE Code for pause key.

X'1033' BUFSTR This word (low - high format) defines the start address of the text buffer. It must point at the first byte of the buffer.

X'1035' BUFLIM This word defines the upper limit of the text buffer (X'9000' for a 32k system).

TEXT BUFFER FORMAT.

The user may find it useful to understand the format of the text buffer which is as follows:-

The first two bytes of the buffer contain the address (low - high) of the end of the buffer. This word is followed by a start of buffer marker, a byte of X'FF'. The end of the buffer has a similar mark. The first character of each line of text has bit 7 set which acts as a delimiter, this byte is normally a space but is also used for line markers.

Note - when Wordease is first entered a test is made by checking that the buffer end address points at a valid end of buffer mark (X'FF'), if not Wordease exits immediately. An empty buffer at X'2800' would look like:-

2800 03 28 FF FF